

Microsoft[®] SQL Server[™] 2005 Reporting Services Step by Step

*Stacia Misner; Hitachi
Consulting*

To learn more about this book, visit Microsoft Learning at
<http://www.microsoft.com/mspress/books/9153.aspx>

9780735622500
Publication Date: April 2006

Microsoft
Press

Table of Contents

Introduction	xiii
Finding Your Best Starting Point	xiii
About the Companion CD-ROM	xiv
System Requirements	xiv
Installing and Using the Sample Files	xv
Conventions and Features in This Book	xvii

Part I Getting Started with Reporting Services

1 Understanding Reporting	3
Reporting Scenarios	3
Reporting User Communities	5
The Enterprise Reporting Life Cycle	6
Authoring	7
Management	7
Access and Delivery	8
Reporting Services Components	8
Authoring Components	9
Management Components	11
Access and Delivery Components	13
Chapter 1 Quick Reference	16
2 Installing Reporting Services	17
Considering Deployment Options	17
Choosing a Reporting Services Edition	17
Planning a Site Configuration	18
Deciding Naming Conventions	19
Preparing for Installation	19
Reviewing Operating System Requirements	19
Reviewing Software Requirements	20

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

	Reviewing Configuration Prerequisites	20
	Creating Reporting Services Credentials	21
	Installing Reporting Services	23
	Launching Setup	23
	Choosing the Components	24
	Selecting the Service Account	27
	Using the Reporting Services Configuration Manager	28
	Configuring Virtual Directories	29
	Specifying the Report Server Database	31
	Specifying Report Delivery Options	32
	Verifying the Installation	34
	Chapter 2 Quick Reference	36
3	Building Your First Report	37
	Authoring a Report	37
	Starting a New Report	38
	Connecting to a Data Source	40
	Getting Data for the Report	43
	Structuring Data in the Report	45
	Placing Data in the Report Structure	46
	Applying a Style Template	48
	Finishing the Report Wizard	49
	Checking the Report Layout	51
	Correcting Report Layout Issues	55
	Publishing a Report	57
	Managing a Report	59
	Reviewing Report Properties	59
	Changing Report Properties	62
	Reviewing Execution Properties	63
	Changing Data Sources Properties	65
	Changing Execution Properties	66
	Accessing a Report	67
	Displaying a Report	67
	Searching a Report	68
	Exporting a Report	69
	Chapter 3 Quick Reference	71

Part II Authoring Reports

4	Developing Basic Reports	75
	Understanding a Report Definition File	75
	Preparing a Report Using Report Designer	76
	Creating a New Report Project	76
	Creating a New Report	77
	Connecting to a Data Source	78
	Working with Datasets	79
	Structuring a Report Using Report Designer	82
	Adding Items from the Toolbox	82
	Working with Table Rows	83
	Sorting Table Rows	86
	Grouping Data in a Table	88
	Adding Group Headers	91
	Computing Group Subtotals	92
	Formatting a Report Using Report Designer	93
	Setting the <i>Format</i> Property	93
	Applying Styles	94
	Editing Properties	96
	Adding Floating Headers	98
	Triggering Page Breaks	100
	Adding a Textbox	100
	Setting Table Properties	103
	Working with Page Headers	104
	Working with Page Footers	105
	Adding Graphical Elements	106
	Adding Images	108
	Chapter 4 Quick Reference	111
5	Working with Expressions	113
	Using Expressions to Calculate Values	113
	Creating Calculated Fields	114
	Using Global Variables	119
	Using the <i>ReportItems</i> Collection	122

- Using Aggregate Functions 125
 - Using Aggregate Functions in a Table 127
 - Using Aggregate Functions in a Textbox 129
- Using Expressions to Change an Object's Behavior 132
 - Using Conditional Formatting 132
 - Sorting 133
- Chapter 5 Quick Reference 135
- 6 Organizing Data in Reports 137**
 - Understanding Data Regions 137
 - Comparing Types of Data Regions 137
 - Using Data Regions 138
 - Using a Matrix 138
 - Adding a Matrix Data Region 139
 - Grouping Rows 140
 - Sorting Rows 141
 - Grouping Columns 142
 - Using Subtotals in a Matrix 144
 - Using Static Rows and Columns in a Matrix 146
 - Using a Chart 148
 - Adding a Chart 148
 - Adding Values and Column Groups to a Chart 149
 - Grouping Data by Series 150
 - Adding a Chart Legend 152
 - Setting Chart Properties 154
 - Using a List 155
 - Adding a List 156
 - Grouping and Sorting a List 157
 - Using Fields in a List 157
 - Nesting Data Regions 158
 - Chapter 6 Quick Reference 160
- 7 Building Advanced Reports 161**
 - Using Parameters to Change Report Data 161
 - Adding a Report Parameter 162

Using the <i>Parameters</i> Collection	164
Adding a Query Parameter	167
Supplying Values for a Query Parameter	169
Creating a Report Parameter for a Filter	172
Adding a Filter	175
Adding a Parameter Value to a Report	177
Linking Information with Interactive Features	177
Adding Actions	179
Using the <i>Hidden</i> Property	182
Using the <i>ToggleItem</i> Property	183
Adding a Subreport	186
Working with Hierarchical Data	188
Displaying a Recursive Hierarchy in a Data Region	189
Using the <i>Level</i> Function	191
Using the <i>Recursive</i> Keyword	192
Creating an Analysis Services Data Source and Dataset	194
Using an Analysis Services Dataset	196
Chapter 7 Quick Reference	197
8 Building Report Models	199
Understanding Report Models	199
Creating a Report Model	200
Adding a Report Model Data Source	200
Adding a Data Source View	201
Adding a Report Model	204
Reviewing a Report Model	207
Modifying a Report Model	211
Changing Model Objects	212
Organizing Model Objects	215
Adding Source Fields	220
Adding Expressions	222
Managing a Report Model	223
Publishing a Report Model	224
Securing a Report Model	226
Chapter 8 Quick Reference	229

Part III Managing the Report Server

9	Managing Content.	233
	Publishing Reports.	233
	Defining Deployment Properties by Project	234
	Deploying Reports.	236
	Uploading Reports.	239
	Creating a Script File	240
	Using the <i>rs</i> Utility.	241
	Organizing Content.	242
	Working with Folders	242
	Moving Content.	244
	Linking Reports	246
	Working with General Properties	248
	Using Properties to Manage Report Content	250
	Working with Data Sources	250
	Specifying a Default Value	253
	Disabling a Parameter Prompt	255
	Managing Report Execution.	256
	Executing Reports with Current Data	257
	Implementing Cached Instances.	259
	Working with Shared Schedules	261
	Managing Snapshots.	263
	Using Report History.	266
	Chapter 9 Quick Reference.	268
10	Managing Security.	271
	Using Report Server Security	271
	Adding Role Assignments	272
	Applying Item Security	277
	Applying System Security.	280
	Applying Data Security	285
	Using Roles and Parameters to Restrict Data	285
	Restricting the Source Query by User.	288
	Filtering the Report Data by User	293
	Chapter 10 Quick Reference	296

11	Managing Server Components	299
	Configuring Reporting Services	299
	Editing the rsreportserver.config File	300
	Changing Encrypted Configuration Information Using the <i>rsconfig</i> Utility	302
	Configuring Tracing on the Report Server	305
	Managing the Report Server	307
	How to Monitor Performance	308
	Managing Execution Logging	308
	Initializing an Execution Log Database	309
	Using an Execution Log Report	312
	Adding Current Data to the Execution Log	315
	Applying Time-outs to Source Queries	317
	Applying Time-outs to Report Execution	318
	Canceling Jobs	320
	Disabling a Shared Data Source	321
	Administering Reporting Services Databases	324
	Monitoring Database Storage Consumption	324
	Implementing a Backup and Restore Strategy	326
	Chapter 11 Quick Reference	328

Part IV Delivering Reports

12	Accessing Reports	333
	Finding and Viewing Reports	333
	Navigating the Folder Hierarchy	334
	Refreshing Reports	336
	Searching for Reports	337
	Using the HTML Viewer	339
	Using My Reports	342
	Saving Reports for Future Reference	346
	Creating a Report History Snapshot	347
	Saving Reports to Local Files	349
	Printing Reports	351
	Chapter 12 Quick Reference	353

13	Rendering Reports	355
	Comparing Rendering Formats	355
	Rendering for Online Viewing	356
	Rendering as MHTML	356
	Rendering for Document Management	358
	Rendering as a TIFF File	358
	Rendering as a PDF File	362
	Rendering for Data Exchange	367
	Rendering as an Excel File	367
	Rendering a Report as a CSV File	371
	Rendering a Report as an XML File	373
	Chapter 13 Quick Reference	374
14	Managing Subscriptions	375
	Creating a Standard Subscription	375
	Delivering a Report by E-Mail	376
	Delivering a Report to a File Share	383
	Creating a Data-Driven Subscription	389
	Creating a Subscription Delivery Table	389
	Creating a Data-Driven Subscription	390
	Managing Subscriptions	398
	Using the My Subscriptions Page	399
	Deleting Subscriptions	403
	Chapter 14 Quick Reference	404
15	Creating Reports with Report Builder	405
	Building Basic Reports	405
	Getting Started with Report Builder	405
	Building a Report	408
	Building a Matrix Report	412
	Using Clickthrough	415
	Building a Chart Report	417
	Enhancing Reports	420
	Formatting a Report	421

Filtering a Report	423
Adding Expressions	426
Chapter 15 Quick Reference	428

Part V Programming Reporting Services

16 Report Authoring with Custom Development	433
Using Custom Code	433
Adding Custom Code to a Report	434
Accessing Functions Using the <i>Code</i> Global Member	436
Creating a Custom Class Library	438
Using Functions from a Custom Assembly	441
Creating Custom Data Processing Extensions	446
Assigning a Namespace	447
Creating a Connection Object	448
Creating a Command Object	450
Creating a Data Reader Object	451
Deploying a Custom Data Processing Extension	454
Generating Report Definition Language	462
Creating a Console Application	462
Adding a Data Source Connection	464
Generating a Fields List	465
Generating the RDL	466
Chapter 16 Quick Reference	469
17 Programming Report Server Management	471
Understanding Web Services	471
Using the <i>rs</i> Utility	472
Querying Report Server	474
Using a Custom Application to Manage Reporting Services	477
Querying Report Server	478
Managing Reports	485
Using the Reporting Services WMI Provider	493
Querying a Report Server	493
Chapter 17 Quick Reference	497

18	Building Custom Reporting Tools	499
	Using URLs	499
	How to Use URLs to Access Reports	500
	Viewing Reports	500
	Working with Report Parameters	503
	Using URL Access Parameters	505
	Using the Web Service	510
	Rendering a Report	511
	Authenticating Users	515
	Using Report Parameters in a Control	517
	Enabling Multiselect for Parameters	524
	Chapter 18 Quick Reference	527
	Glossary	529
	Index	531

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

Chapter 1

Understanding Reporting

After completing this chapter, you will be able to:

- Understand the purpose of enterprise and ad hoc reporting.
- Recognize the characteristics of a reporting platform.
- List the constituents of reporting user communities.
- Describe the stages of the reporting life cycle.
- List the features and components of Microsoft SQL Server 2005 Reporting Services.

In this chapter, you see what enterprise reporting and ad hoc reporting are all about, as well as how they differ from other types of reporting. You also review how different groups within your organization need to use or support reporting and how they participate in the reporting life cycle. With this foundation, you'll better understand how the various components of Reporting Services fully support the reporting needs of your organization.

Reporting Scenarios

Since you're reading this book, it's likely that you work for a company that needs to be able to share information. Whether your company is small or large, can you imagine what would happen if employees couldn't access the information they need to do their jobs? The decisions that each individual employee makes during the course of his daily tasks have a profound impact on successful business operations and rely on easy and regular access to information.

One way that a company commonly shares information is through enterprise reports. For the purposes of this book, an *enterprise report* is considered to be the presentation of information that is formally distributed to some or all individuals across an enterprise, or even to individuals outside the enterprise. This information can be presented in a variety of formats, for example, as a Microsoft Excel spreadsheet or a text document. It can also be delivered as a printed report or sent to a list of recipients as an e-mail attachment. Information can also be made available in a central location, such as on a Web page on a corporate intranet or embedded in a portal where users can access reports when needed.

Sometimes a company needs a less formal way of creating and sharing information, especially if business dynamics are changing rapidly and there's a need to provide quick access to current information to a small group. In this situation, it's less important to adhere to presentation standards, yet the quality of information must be consistent with traditional enterprise reports. These *ad hoc reports* tend to be simpler than enterprise reports in terms of data content and

presentation features, and are usually based on standard nontechnical representations, or *models*, of underlying data sources. A business person should be able to create an ad hoc report without knowing how to write a relational query and without having to wait for technical assistance in order to pull together data to answer a new business question.

The diverse types of organizations that use reporting and the differences in their information needs make it difficult to compile a comprehensive list of all possible reporting scenarios. However, you can look at who is using the shared information to develop the following generalizations about reporting scenarios:

- *Internal reporting* is probably the most common enterprise reporting. This category of reporting involves the sharing of information within an organization across all levels of employees and usually involves standard departmental reports. For example, employees in the product warehouse might regularly receive detailed printed order reports every morning. Elsewhere in the company, managers might get a financial statement in an Excel workbook delivered as an e-mail attachment when the books are closed each month.
- *External reporting* can take many forms, but is defined as disseminating information to people outside an organization. This information might be printed and mailed, such as shareholder reports. Increasingly, companies are publishing annual reports as PDFs (Portable Document Format files) for interested parties. External reporting can even include the exchange of information between business systems, such as invoicing information sent to a customer's receivables system electronically.

You can also consider how the information is being accessed to develop the following additional generalizations:

- *Standard reporting* relies on a central storage location that can display a list of contents or a catalog of the available reports so users can find the reports they need. Usually, security is applied to report storage to control the reports that individual users can see. Reports might be organized in a proprietary reporting platform repository or some other type of document management system.
- *Ad hoc reporting* depends on the availability of a model that allows users to select which data elements should be included in a report, along with a designer tool that allows them to arrange the layout of these elements to produce a simple report. These reports might be saved to a central repository to share with others or they might be stored on the user's local hard drive for personal reference.
- *Embedded reporting* is the integration of reports into portals and in-house or third-party applications. For example, many companies are migrating to Web-based line-of-business applications for accounting and payroll functions. Instead of building reporting processes into these applications, these companies can leverage an extensible enterprise reporting platform to allow users to access information by using these applications.

These reporting scenarios have the following two characteristics in common:

- **Central storage** Reports or report models (for ad hoc reporting) are accessed from a central location. Reports might also be delivered directly to users from a central server. Many people need access to the same information, possibly in different formats. Often, access to information needs to be limited to those with a need to know.
- **Standardization** Enterprise reports conform to a standard design with a consistent layout, while ad hoc reports conform to a standard data model to ensure consistent results.

In addition, the proliferation of information that can (or should) be available to the average worker has led to increasingly more sophisticated requirements for an enterprise reporting solution. For example, users need to be able to do the following:

- Navigate easily within a large report.
- Move from one report to another while maintaining context.
- Access previous versions of a report to compare information at different points in time.
- View data consolidated from multiple sources into a single report.

An enterprise reporting solution also needs to satisfy administrative requirements. A reporting platform should have the following characteristics:

- Flexibility to store a single report from which multiple versions may be generated based on changeable parameters or user profiles.
- Ability to support a push-pull paradigm, in which users can seek out the information they need online or subscribe to information that is sent to them on a periodic basis.
- Capability to manage reports using a Web interface so that administrators can perform tasks without being tied to their desks.

Reporting User Communities

Many people within an organization are usually involved in some aspect of reporting. Typically, users are members of one or more of the following communities: information consumers, information explorers, and analysts.

Most users—typically 65–80 percent of the total user population—are information consumers. Information consumers usually view static and predefined reports. If they use printed reports, they might get them the old-fashioned way—someone does a batch print of hard copy reports, then sends it out or delivers it to each recipient’s in-box. A more technically oriented environment might make a document repository available, providing the electronic equivalent of a file cabinet that information consumers can access at will. In some cases, information consumers need to receive reports on a recurring basis, such as a weekly update on key performance measurements. Some of these users might want to produce their own simpler reports when an existing report doesn’t answer a particular question.

One of the many strengths of Reporting Services is its ability to provide easy access to a wide array of predefined reports, making information consumers a key audience served by Reporting Services. Although many people might prefer to view information online, they can still get printed reports or can get reports delivered via e-mail. In either case, reports can be processed on demand (where information is as current as the data in the source system) or on a scheduled basis (where information represents a specific point in time). For maximum flexibility, an information consumer can choose from a variety of formats that can be delivered to a range of devices. Finally, Reporting Services provides ad hoc reporting capabilities so information consumers can obtain quick answers as needed.

Information explorers typically constitute 15–25 percent of the user population. Like information consumers, they use predefined reports, but they also interact with reports. For example, information explorers commonly use filters to isolate segments of data. Information explorers might also interact with reports by starting with summary information and then moving to more granular levels of detail, whether drilling down to view details in the same report or drilling across to view related information in a separate report.

Interactive reports suitable for information explorers require more work to develop than static reports, but Reporting Services has a wide array of features to support the development of these reports. Parameters can be designed into a report to support filtering data at the source or in the report. An information explorer can change parameter values on demand, or an administrator can predefine specific parameters for different groups of information explorers. Reports can also include dynamic visibility to support drilling down or actions to support drilling across.

The smallest user community, typically representing 5–10 percent of users, includes analysts. This group possesses the skills to develop free-form reports that facilitate complex data analysis. Such reports are often in spreadsheet form; through them, analysts can enhance the data with sophisticated calculations, such as linear regressions and allocations. These reports can eventually be shared with information consumers and information explorers.

Out of the box, Reporting Services supports analytical needs by providing the ability to export a report to Excel. Conversely, an Excel workbook created by an analyst can be uploaded to Reporting Services as a resource to be shared with the rest of the user community. Also, because Reporting Services is an extensible architecture, a custom application or third-party plug-in can give analysts complete flexibility to develop free-form reports within the Reporting Services environment.

The Enterprise Reporting Life Cycle

The *enterprise reporting life cycle* is a three-stage process through which a report progresses from authoring to management to delivery. A reporting platform must not only serve the needs of each reporting community, but must also fully support each stage of the reporting life cycle. It should also provide the architecture, functionality, and utilities to support the

activities of authoring, managing, and delivering reports. In other words, everything you need from the beginning to the end of the reporting process should be in one integrated product set. Reporting Services provides just that.

Authoring

The primary activities of the authoring stage include defining the data to be presented in a report, organizing the data into a structured layout, and applying formatting to enhance the report's appearance. For example, when executive management needs to monitor sales performance across product lines, a report author can create either an enterprise report or an ad hoc report to present sales data in a table layout. To facilitate analysis, however, the report author might use conditional formatting to highlight products for which performance exceeds defined performance goals or fails to meet these goals. This advanced formatting technique is an enterprise reporting activity.

To support authoring, Reporting Services provides a broad set of features to present data in structures, such as tables and charts, to group data within these structures, to allow calculations, and to add formatting. This reporting platform also facilitates access to a variety of organizational data sources, such as online transaction processing (OLTP) systems or data warehouses that store relational or online analytical processing (OLAP) data. Reporting Services allows an enterprise report author to easily combine data from multiple sources into a single report. All types of structured data are supported for enterprise reports—relational, hierarchical, and multidimensional data. To access data sources not explicitly supported by Reporting Services, custom data processing extensions can be added, which means the possibilities are endless.

In addition, Reporting Services allows the report author to design a report with consideration for its purpose. An ad hoc report author can add interactive sorting and apply filtering to a report, while an enterprise report author has access to a rich feature set that enables the development of both static and interactive reports for the full spectrum of the user community. For example, static enterprise reports, such as print-ready invoices for mailing, or interactive online reports, such as the presentation of key performance indicators accessible from the corporate portal, can be quickly and easily developed. Interactive reports that factor in how users need to explore and analyze data can also be designed. Parameters, dynamic visibility, and actions can be used individually or in combination to affect both the information visible in an enterprise report and the information's appearance.

Management

The management stage begins when a report or ad hoc reporting model is published to a server. This stage continues with the organization of the report or model with other content on the server and the performance of other administrative tasks, such as setting report properties, managing report execution, and applying security. Either a report author or an administrator is responsible for publishing a report to a centrally managed server. When the report

is on the server, a report execution schedule can be established to update the report regularly, such as every Monday morning. In addition, security is applied to the report so that only certain users, perhaps executive management, can view the report.

Reporting Services provides mechanisms to publish reports and models to a central server through the authoring tool or through management tools. Once a report is online, security can be implemented to control access. Further, the execution of reports is configurable so that reports can be produced on demand or on a scheduled basis. Reporting Services includes all this functionality using a server infrastructure that can exist on a single server or be distributed across many servers or incorporated into a Web farm.

Access and Delivery

The access and delivery stage includes all activities related to the distribution of reports from a central repository, such as accessing reports online, rendering reports to various formats, saving and printing reports, and subscribing to reports. Some users, for instance, might choose to receive reports via e-mail as soon as these reports are executed each week or to receive a notification that a report is available for viewing online. Other users, by contrast, might prefer to view the reports online using the company intranet only as the need for information arises.

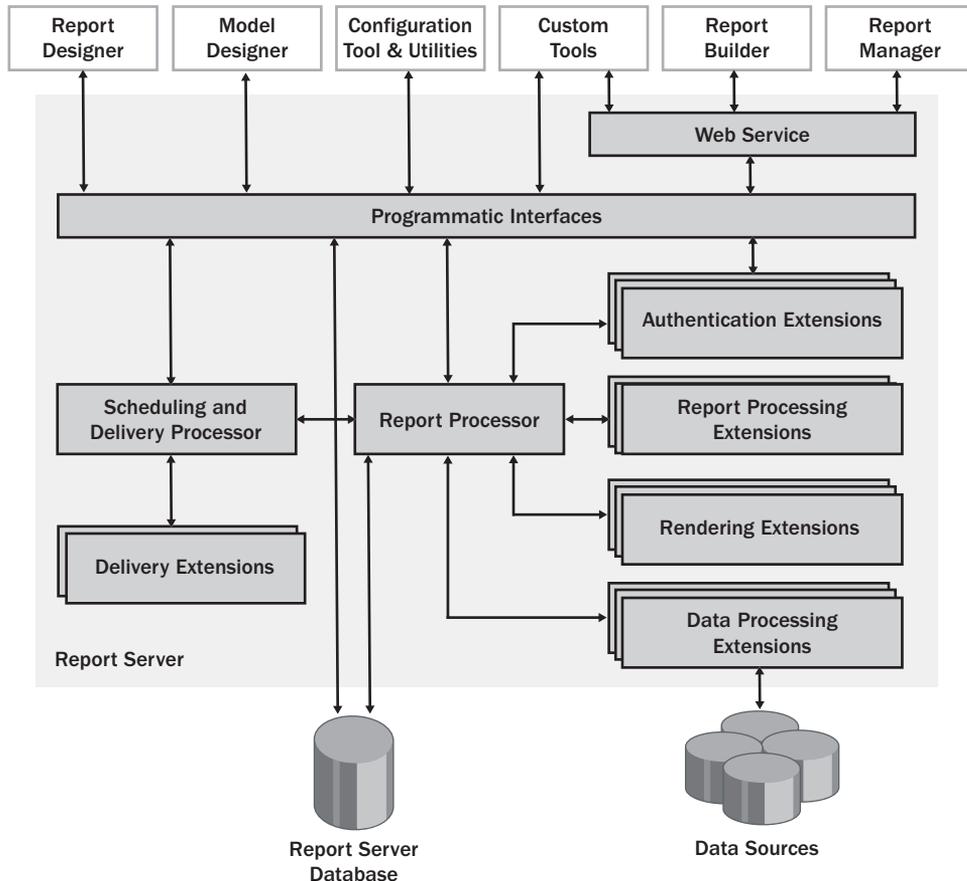
To support delivery, Reporting Services can produce reports using a variety of output formats which are referred to as rendering formats. A report can be made available through the intranet using a Web browser, or it can be sent embedded in an e-mail message or as an e-mail attachment in many formats, such as PDF or Excel. Reporting Services also provides flexible delivery mechanisms to support both push and pull distribution methods for internal and external users. Because Reporting Services is an extensible system, you can add rendering formats, different security frameworks, or alternative delivery options. In addition, the access or delivery of reports can be integrated into corporate applications.

Reporting Services Components

The requirements of a solid enterprise reporting solution are formidable, but Reporting Services meets this challenge with a set of integrated, multitiered components. Because Reporting Services is a Microsoft .NET-based platform that can use both a Web service and an application programming interface (API), it can be customized to fit within existing technical infrastructures. Furthermore, by separating components into discrete functional units, the Reporting Services architecture can be scaled to accommodate even the largest organization by distributing components across several servers. (You learn how to install these components in Chapter 2, “Installing Reporting Services.”) Together, these components support the authoring, management, access, and delivery requirements of a reporting platform.

The Reporting Services architecture consists of three layers. The application layer includes two different client components for authoring—one to author enterprise reports, and the other

to author ad hoc reports. The application layer also includes a client component for managing one or more Report Servers within a single server management interface, and a server component called Report Manager, which is installed on a Web server and used for report access and for some server management tasks. The server layer is the Report Server where all the processing and management of the reporting platform occurs. The data layer includes data providers to access data sources used in reports, as well as a pair of databases for the storage of reports and information used by Report Server. These components can be installed on a single server or distributed across several servers. The Reporting Services components are illustrated in the following figure:



Authoring Components

When Reporting Services is installed, a client component called Report Designer is added as a set of templates to the Microsoft Visual Studio 2005 development environment. If you don't have this version of Visual Studio installed, installation of Reporting Services will also install the requisite components so you can use Report Designer.

The easiest way to build enterprise reports is to use the Report Designer templates. If you're a report author, you don't need to have programming skills to effectively use this tool. However, if you're already an experienced programmer, you can also take advantage of the programmatic interfaces to build a custom authoring tool.

Report Builder is a thin client application that installs on your computer when you first launch this application through Report Manager or by accessing its URL on the Report Server. Report Builder is used for ad hoc reporting and, like Report Designer, does not require programming skills to create useful reports. In fact, you don't even need to know how to write a query to access data, since the query is written for you based on the items you select from the report model, which is created using Model Designer, another client component that is added to the Visual Studio environment.

Report Designer

As an enterprise report author, you can use a graphical interface to build feature-rich reports using drag-and-drop techniques to create a query to retrieve data and to define the layout and appearance of data in the report. You can use ActiveX Data Objects (ADO) .NET-managed data providers to access many OLE DB and Open Database Connectivity (ODBC) data sources. If you need access to other data sources, you can build your own data providers. After defining a query to retrieve data from a selected data source, you use Report Designer to place data into one or more structures. You also use Report Designer to apply calculations to the data, as well as to access a complete set of features that support presentation options such as formatting and visibility. The result is a report definition in the form of an XML document using a nonproprietary schema known as Report Definition Language (RDL). You learn about Report Designer and Report Definition Language in Chapter 4, "Developing Basic Reports."

Report Designer also includes a preview version of the functionality used by the server to produce reports, so you can test a report before putting it on the server. That way, you can get an idea of how the users will see the report while you're still developing it. When you're ready to publish the report to the server, you use the Visual Studio build and deploy processes.

Report Builder

If you're an ad hoc report author, you employ a separate graphical interface to build simple reports by using drag-and-drop techniques to retrieve data from a single source and to define the layout and appearance of data in the report. Instead of defining a data source and building a query, as you do with Report Designer, you select a report model, which contains the data connection, metadata, and data relationships needed to produce and execute a query. In the current release of Report Builder, a report model can only access relational data in a SQL Server database. If you use the Standard, Enterprise, or Developer Edition, you can also access multi-dimensional data in an Analysis Services database. After selecting a report model, you select a table, matrix, or chart template, which becomes the structure into which you place data fields

defined by the report model. You can also use Report Builder to add your own calculations, sort or filter data, and apply formatting, such as font colors or styles. You can view the report within Report Builder and, if you want to share it with other users of Reporting Services, you can publish the report to the server. You learn about Report Builder in Chapter 15, “Creating Reports with Report Builder.”

Model Designer

The report model on which Report Builder depends is created using the Model Designer interface. You begin development of the report model by specifying a data source and specifying which tables and their relationships to use from that data source. Then you use a wizard to generate a report model, which will become the business description of the underlying data. You can make changes to the names of objects generated in the model, rearrange and organize these objects to simplify the users’ navigation in the model, and remove unneeded objects. You learn about Report Builder in Chapter 8, “Building Report Models.”

Programmatic Interface for Authoring

Using the Reporting Services API, application developers can build custom applications to create reports or add functionality to reports. In Chapter 16, “Report Authoring with Custom Development,” you learn how to use custom code to extend the authoring capabilities of Reporting Services and how to generate RDL files programmatically.

Management Components

After you’ve installed Reporting Services, you can use the Reporting Services Configuration tool to set or modify configuration settings. The Report Manager is installed on a Web server and used for both management and for access and delivery tasks. Report management capabilities are also accessible in Microsoft SQL Server Management Studio. All of the tasks you can perform using Report Manager can also be performed within this management component if it’s installed on your local computer. Additionally, Reporting Services provides command-line utilities for specific server management tasks. You also have the option to build your own Windows or Web-based management tools using the Reporting Services API.

Reporting Services Configuration Tool

The Reporting Services Configuration tool allows you to configure local or remote instances of Reporting Services. This tool is a client application you can employ to manage virtual directories used by the report server and Report Manager, define Windows accounts to run the Web and Windows services, create the report server database, manage encryption keys to protect report server data, and configure Simple Mail Transfer Protocol (SMTP) settings. Typically, you configure these settings once after installation, but as your reporting environment changes, you can use this tool to update the report server configuration.

Report Manager

Report Manager is a Web-based content management tool included with Reporting Services. Reporting Services separates administrative tasks into two main groups: content management and system management. If you're an administrator responsible for content management, you can use Report Manager to manage how reports are organized on the Report Server and how users interact with those reports. Content management using Report Manager is covered in more detail in Chapter 9, "Managing Content." As an administrator responsible for server resources and performance, you can use Report Manager to configure execution options (described in Chapter 9), to set security (described in Chapter 10, "Managing Security"), and to manage subscription and delivery options (described in Chapter 14, "Managing Subscriptions").

Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio is a new workstation component provided with Microsoft SQL Server 2005 as a single environment from which to administer any of the server components of SQL Server. This environment does not replace Report Manager, nor does it provide additional features. Instead, it gives you another way to perform the same activities. For example, you can manage content, configure security, and set options for report execution or subscriptions from the same management interface that you use to manage SQL Server or Analysis Services if you're responsible for administering these other server types. This interface can also be useful if you manage multiple report servers.

Command-Line Utilities

Reporting Services includes command-line utilities that allow you to manage a Report Server locally or from a remote location. The following command-line utilities are provided by Reporting Services for server administrators:

- **rsconfig** A connection management utility that can change the connection used by Report Server to connect to the ReportServer database.
- **rs** A script host that you can use to execute Microsoft Visual Basic .NET scripts for management tasks, such as publishing reports or copying data between ReportServer databases.
- **rskeymgmt** An encryption key management tool that you can use to back up encryption keys for future recovery of a database or to change encrypted data used by a Report Server.

You learn more about these utilities in Chapter 11, "Managing Server Components," except for the *rs* utility, which is discussed in Chapter 17, "Programming Report Server Management."

Programmatic Interface for Management

You can also use the Reporting Services API to perform server management activities, such as publishing or deleting reports. You can build your own application, or you can build Visual Basic .NET scripts to use in conjunction with the *rs* utility to perform administrative tasks on Report Server. You learn how to use the Reporting Services API for server management in Chapter 17.

Access and Delivery Components

The components of Reporting Services that are involved in the access and delivery of reports break down into two groups: client components and server components. You can choose to use the supplied client components or build your own client applications. However, you must use the core server components of Reporting Services, although you can use custom or third-party applications to extend the capabilities of the server components.

Report Manager

Report Manager is not just a management tool. The user community also uses Report Manager to access reports and subscribe to reports. You learn about general report access using Report Manager in Chapter 12, “Accessing Reports.”

Processor Components

Report Server is the heart of Reporting Services. Although administrators interact with Report Server using the management components, the bulk of activity that occurs on Report Server is related to supporting the access and delivery of reports. Report Server runs as a Web service, allowing Report Manager, your own custom programs, and third-party applications to access server processes.

Report Server uses Microsoft Internet Information Services (IIS) to receive requests, and then activates the applicable subcomponents in response to the requests. There are two processor subcomponents of Report Server that act as command central to manage these requests and the corresponding output returned by the other subcomponents. The core processor, Report Processor, handles all requests related to the execution of reports and the production of the final output. To complete these requests, the Report Processor calls other subcomponents, referred to as *extensions*, to handle data processing, rendering, and security. The Scheduling and Delivery Processor responds to scheduled events and delivers reports. This processor uses delivery extensions to send reports to their destinations.

Report Processor The Report Processor is responsible for retrieving the report layout from the report definition and merging it with data returned from the query included in the report definition. At this point, the report is in an *intermediate format*. What happens next depends on the report request. If a user wants to see the report online, the intermediate format is

passed to the appropriate rendering extension so the proper output format can be created, such as a Web page.

On the other hand, if the request is to generate a *report snapshot* (which is a report at a specific point in time), the intermediate format is stored in the ReportServer database. When a user requests the snapshot, the Report Processor retrieves the intermediate format, calls the rendering extension, and then sends the final format of the report to the user.

The Report Processor also manages requests for report models. These requests occur when a user browses a report model or runs a report in Report Builder, including drillthrough reports.

By separating the presentation processing from data retrieval and rendering, multiple users can view the same report at the same time, and each can change the viewing format. You learn more about execution options in Chapter 9.

Scheduling and Delivery Processor As its name implies, the Scheduling and Delivery Processor is responsible for running scheduled reports and for delivering reports to a location or a device on a scheduled basis. It uses SQL Server Agent to process schedules. When the applicable time arrives, SQL Server Agent sends instructions related to the schedule to the Scheduling and Delivery Processor. The report is rendered and passed to the delivery extension to send the report to specified recipients or a target location on a file share.

A report snapshot is an example of a scheduled report. The schedule information is specified by a user or administrator using Report Manager and stored in the ReportServer database. When the Scheduling and Delivery Processor finishes processing the snapshot, the intermediate format is stored in the ReportServer database. When a user wants to view the report, Report Server takes over by retrieving the intermediate format of the snapshot and calling the specified rendering extension to finalize the output. You learn more about scheduling reports in Chapter 9.

When users subscribe to reports, the desired delivery extension is selected and the time of delivery is specified. When the time comes to deliver a report, the Scheduling and Delivery Processor gets a rendered report and then passes the report to the applicable delivery extension. Delivery options for subscriptions are described in Chapter 14.

Server Extensions

Server extensions are used to perform specific functions. Reporting Services uses five different types of server extensions: authentication, data processing, report processing, rendering, and delivery. Over time, you can expect to see more extensions available for Reporting Services, whether developed by Microsoft or by other commercial software developers.

Authentication Extensions Authentication extensions are used to define the authorization model used by Reporting Services. Only one authentication extension is supplied, which supports Microsoft Windows and SQL Server security. You can, of course, create your own authentication extension to integrate Reporting Services with another security architecture.

Data Processing Extensions Data processing extensions are responsible for processing the query requests received from the Report Processor. The query request includes a data source, a query, and possibly, query parameters. The applicable data processing extension then opens a connection to the data source, returns a list of field names from the query, executes the query, and retrieves the query results, which are then returned to the Report Processor.

Reporting Services comes with six data processing extensions: SQL Server, Analysis Services, Oracle, OLE DB, ODBC, and XML. However, you can also use any ADO.NET data provider or build your own data processing extensions. You learn how to create a custom data processing extension in Chapter 16.

Report Processing Extensions Report processing extensions are used to process custom report items that may be embedded in a report. The Report Server already knows how to process standard report items, such as tables and text boxes, but if you add a custom report item—like a special gauge control or an embedded map from Microsoft MapPoint—then you need to provide an extension to handle the custom processing required by the new report item.

Rendering Extensions Rendering extensions are called by the Report Processor to take the data that was received from the data processing extension and merge that data with the report definition. The result is a finished report in a format specific to the device that will receive the report.

At the time of this writing, Reporting Services has the following rendering extensions: HTML, MHTML (MIME Encapsulation HTML), Excel, Acrobat PDF, CSV (comma separated values), and XML. As with other extension types, you can develop your own rendering extension to produce other output formats.

Delivery Extensions Reporting Services currently includes the following three delivery extensions:

- The e-mail delivery extension allows Reporting Services to embed a report in an e-mail message or send the report as an attachment. Alternatively, an e-mail notification can be sent that includes a link to the report. If the delivery is an e-mail notification, it can also be sent without the link to a pager, cellular phone, or any device that can receive a simple message.
- The file share delivery extension can be used to store reports in a centrally accessible location independent of the ReportServer database or as part of a report archive strategy.
- A null delivery provider is available for data-driven subscriptions to periodically load reports into the cache in advance of user viewing. This option is useful for reports that take a long time to execute.

You can also develop your own delivery extension to expand the delivery functionality of Reporting Services.

ReportServer Databases

Reporting Services centralizes report storage in two SQL Server databases. The ReportServer database stores information used to manage reports and resources, along with the reports themselves. In addition, this database is the storage location for security settings, encrypted data, data related to schedules and delivery, and information about extensions. The ReportServerTempDB database stores temporary data used for caching purposes. More information about these databases can be found in Chapter 11.

Programmatic Interface for Access and Delivery

You can use the Reporting Services API to create assemblies when you need to accommodate specialized security, data processing, rendering, or delivery scenarios. In Chapter 16, you learn how to build a custom data processing extension. The Reporting Services API also enables you to develop your own applications to allow users to view reports or to produce reports using different formats. You learn more about custom reporting in Chapter 18, “Building Custom Reporting Tools.”

Chapter 1 Quick Reference

This term	Means this
Report	Information that is structured and formatted for print or online viewing.
Enterprise reporting	Sharing of information on a regular basis across a wide audience.
Ad hoc reporting	Reviewing, and possibly sharing, limited information with limited formatting requirements on an as-needed basis.
Enterprise reporting life cycle	The process of authoring, managing, and accessing reports.
Extensions	Subcomponents of Report Server used to provide specific functionality, such as data processing, rendering, report processing, authentication, and delivery.
Intermediate format	The result of merging data from a query with layout information from a report definition. The intermediate format is sent to a rendering extension to produce the final output, such as a Web page or an Excel file.
Report snapshot	A report that preserves a record of data at a point in time. A report snapshot is stored in the ReportServer database in its intermediate format and rendered only when a user requests the report.